Comparison between different types of $\mu Controller$:

1- Arduino UNO
2- ESP32
3- RPI Pico Series
4- ESP32-S

$\mu Controller$:	Core	RAM	Wi-Fi	GPIO	Real-Time	Power Consumption	Prog. Language	Cost
Arduino UNO ATMega328P	1	2 KB	❌	~20	Limited	Low	Arduino-IDE, C	10$
ESP32	2	520 KB	✅	~30	FreeRTOS	Moderate	ESP-IDF Arduino-IDE	5$
ESP32-S	2	520 KB	✅	~30	FreeRTOS	Moderate	ESP-IDF Arduino-IDE	6$
RPI Pico Series	2	264 KB	❌	~26	FreeRTOS	Low	MicroPython C	4$

Conclusion:

After thorough research and comparison, the ESP32 stands out as the most suitable microcontroller for real-time systems involving sensor data collection and cloud communication.

The ESP32 offers the following advantages:

- Dual-core processor: Enables efficient multitasking, allowing one core to handle sensor data acquisition and the other to manage cloud communication or real-time tasks.

- Sufficient memory: Provides ample RAM and flash for handling sensor buffers, real-time processing, and network stacks.

- ESP-IDF support: Offers a robust development framework specifically designed for real-time and embedded applications, supporting FreeRTOS out of the box.

- Connectivity: Built-in Wi-Fi and Bluetooth eliminate the need for additional modules for cloud integration.

- Power efficiency: Operates at low to moderate power consumption levels, making it suitable for both battery-powered and plug-in systems.

- Cost-effective: Despite its rich feature set, the ESP32 is very affordable, making it ideal for both prototyping and production.

Considering all these factors, ESP32 provides the best balance of performance, connectivity, development support, and cost for our real-time IoT system.